Notebook

March 9, 2025

root/finetune/train.ipynb

[1]:

```
import pandas as pd
import json
from datasets import Dataset, concatenate_datasets
from unsloth import FastLanguageModel
from transformers import TrainingArguments, DataCollatorForSeq2Seq
from trl import SFTTrainer
from unsloth.chat_templates import get_chat_template, train_on_responses_only

#   1.   Excel
xlsx_path = "/root/autodl-tmp/result_with_accuracy.xlsx"
df_xlsx = pd.read_excel(xlsx_path, usecols=["Disease", "Symptoms",␣
 ↪"Treatments"])
df_xlsx.fillna("", inplace=True)   #

dataset_xlsx = Dataset.from_pandas(df_xlsx)

#   2.   JSON
json_path = "/root/autodl-tmp/updated_data.json"
with open(json_path, "r", encoding="utf-8") as f:
    json_data = json.load(f)

formatted_qa = []
formatted_textbook = []

for item in json_data:
    if item["type"] == "qa":
        user_input =␣
 ↪f"<|start_header_id|>user<|end_header_id|>\n\n{item['question']}\n\n"
        assistant_response =␣
 ↪f"<|start_header_id|>assistant<|end_header_id|>\n\n{item['answer']}\n\n"
        formatted_qa.append({"text": user_input + assistant_response})

    elif item["type"] == "textbook":
        assistant_response =␣
 ↪f"<|start_header_id|>assistant<|end_header_id|>\n\n{item['text']}\n\n"
        formatted_textbook.append({"text": assistant_response})
```

```
dataset_qa = Dataset.from_list(formatted_qa)
dataset_textbook = Dataset.from_list(formatted_textbook)

#  3.   Excel
def formatting_prompts_func(examples):
    formatted_texts = []
    for disease, symptoms, treatments in zip(examples["Disease"],␣
 ↪examples["Symptoms"], examples["Treatments"]):
        user_input =␣
 ↪f"<|start_header_id|>user<|end_header_id|>\n\n{disease}\n\n"
        assistant_response =␣
 ↪f"<|start_header_id|>assistant<|end_header_id|>\n\n"
        assistant_response += f"Symptoms:\n{symptoms}\n\n" if symptoms else ""
        assistant_response += f"Treatments:\n{treatments}\n\n" if treatments␣
 ↪else ""
        formatted_texts.append(user_input + assistant_response)
    return {"text": formatted_texts}

dataset_xlsx = dataset_xlsx.map(formatting_prompts_func, batched=True)

#  4.
dataset_combined = concatenate_datasets([dataset_xlsx, dataset_qa,␣
 ↪dataset_textbook])

#  5.   `unsloth/Llama-3.1-8B-Instruct-bnb-4bit`
model_name = "/root/autodl-tmp/Llama-3.1-8B-Instruct-bnb-4bit"
max_seq_length = 2048

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name=model_name,
    max_seq_length=max_seq_length,
    load_in_4bit=True,
)

#  6.   `LoRA`
model = FastLanguageModel.get_peft_model(
    model,
    r=8,  #    LoRA     16   8
    target_modules=["q_proj", "v_proj", "o_proj"],  #
    lora_alpha=8,  #    LoRA     16   8
    lora_dropout=0.05,  #    dropout
    bias="none",
    use_gradient_checkpointing="unsloth",
)

#
tokenizer = get_chat_template(tokenizer, chat_template="llama-3.1")
```

```
#   7.
trainer = SFTTrainer(
    model=model,
    tokenizer=tokenizer,
    train_dataset=dataset_combined,
    dataset_text_field="text",
    max_seq_length=max_seq_length,
    data_collator=DataCollatorForSeq2Seq(tokenizer=tokenizer),
    dataset_num_proc=2,
    packing=False,
    args=TrainingArguments(
        per_device_train_batch_size=8,  #     batch size
        gradient_accumulation_steps=1,  #        LoRA
        warmup_steps=10,  #    warmup
        max_steps=500,  #
        learning_rate=5e-5,  #
        fp16=False,
        bf16=True,
        logging_steps=1,
        optim="adamw_8bit",
        weight_decay=0.01,
        lr_scheduler_type="cosine",  #     `cosine`
        seed=3407,
        output_dir="trained_model",
        report_to="none",
    ),
)

#   8.     `assistant`
trainer = train_on_responses_only(
    trainer,
    instruction_part="<|start_header_id|>user<|end_header_id|>\n\n",
    response_part="<|start_header_id|>assistant<|end_header_id|>\n\n",
)

#   9.
trainer.train()

#   10.
trainer.model.save_pretrained("/root/autodl-tmp/trained_model")
trainer.tokenizer.save_pretrained("/root/autodl-tmp/trained_model")
```

```
 Unsloth: Will patch your computer to enable 2x faster free finetuning.
 Unsloth Zoo will now patch everything to make training faster!
```

```
Map:   0%|          | 0/410 [00:00<?, ? examples/s]
```

```
==((====))==  Unsloth 2025.2.5: Fast Llama patching. Transformers: 4.48.3.
```

```
   \\   /|    GPU: NVIDIA GeForce RTX 4090. Max memory: 23.643 GB. Platform:
Linux.
O^O/ \_/ \    Torch: 2.6.0+cu124. CUDA: 8.9. CUDA Toolkit: 12.4. Triton: 3.2.0
\        /    Bfloat16 = TRUE. FA [Xformers = 0.0.29.post2. FA2 = False]
 "-____-"     Free Apache license: http://github.com/unslothai/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which are red
colored!
```

```
Unsloth: Dropout = 0 is supported for fast patching. You are using dropout =
0.05.
Unsloth will patch all other layers, except LoRA matrices, causing a performance
hit.
Unsloth 2025.2.5 patched 32 layers with 0 QKV layers, 0 O layers and 0 MLP
layers.
```

```
Map (num_proc=2):   0%|          | 0/276485 [00:00<?, ? examples/s]
```

```
Detected kernel version 5.4.0, which is below the recommended minimum of 5.5.0;
this can cause the process to hang. It is recommended to upgrade the kernel to
the minimum version or higher.
```

```
Map:   0%|          | 0/276485 [00:00<?, ? examples/s]
```

```
==((====))==  Unsloth - 2x faster free finetuning | Num GPUs = 1
   \\   /|    Num examples = 276,485 | Num Epochs = 1
O^O/ \_/ \    Batch size per device = 8 | Gradient Accumulation steps = 1
\        /    Total batch size = 8 | Total steps = 500
 "-____-"     Number of trainable parameters = 5,505,024
```

```
<IPython.core.display.HTML object>
```

```
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class
instead.
```

```
[1]: ('/root/autodl-tmp/trained_model/tokenizer_config.json',
      '/root/autodl-tmp/trained_model/special_tokens_map.json',
      '/root/autodl-tmp/trained_model/tokenizer.json')
```

This notebook was converted with convert.ploomber.io